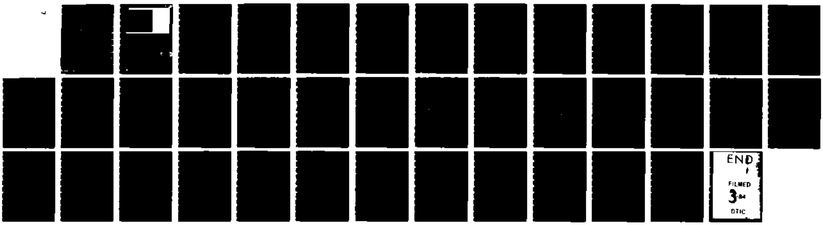
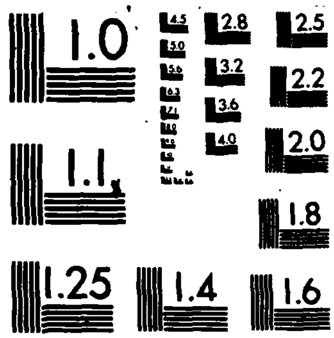
AD-A137 930 A BIVARIATE C2 CLOUGH-TOCHER SCHEME(U) WISCONSIN UNIV-MADISON MATHEMATICS RESEARCH CENTER P ALFELD JAN 84 MRC-TSR-2620 DAAG29-80-C-0041 1/1

UNCLASSIFIED F/G 12/1 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

3

ADA137930

MRC Technical Summary Report #2620

A BIVARIATE C^2 CLOUGH-TOCHER SCHEME

Peter Alfeld

Mathematics Research Center
University of Wisconsin—Madison
610 Walnut Street
Madison, Wisconsin 53705

January 1984

(Received November 28, 1983)

Approved for public release
Distribution unlimited

Sponsored by

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park
North Carolina 27709

U. S. Department of Energy
Washington, DC 20545

-a-

UNIVERSITY OF WISCONSIN-MADISON
MATHEMATICS RESEARCH CENTER

A BIVARIATE C^2 CLOUGH-TOCHER SCHEME

Peter Alfeld*

Technical Summary Report #2620
January 1983

ABSTRACT

A Clough-Tocher like interpolation scheme is described for values of position, gradient and Hessian at scattered points in two variables. The domain is assumed to have been triangulated. The interpolant has local support, is globally twice differentiable, piecewise polynomial, and reproduces polynomials of degree up to three exactly.

AMS (MOS) Subject Classifications: 65D05, 65D07

Key Words: Bivariate interpolation, Clough-Tocher schemes

Work Unit Number 3 (Numerical Analysis and Scientific Computing)

*Department of Mathematics, University of Utah, Salt Lake City, Utah 84112

Sponsored by the United States Army under Contract No. DAAG29-80-C-0041. This research was also supported by the Department of Energy under Contract No. DE-AC02-82ER12046 to the University of Utah and partly by a faculty grant from the University of Utah Research Committee.

SIGNIFICANCE AND EXPLANATION

Some design applications (e.g. in the aircraft industry) require twice differentiable surfaces. In this paper, an interpolation scheme for the construction of such surfaces is derived.

More specifically, the scheme requires values of position, gradient, and Hessian at scattered points in two variables. The domain is assumed to have been triangulated. The interpolant has local support (i.e. evaluation at a point in a specific triangle requires data only on that triangle), is globally twice differentiable, piecewise polynomial, and reproduces polynomials of degree up to three exactly. Explicit formulas are given for the coefficients of the interpolant. A pilot (FORTRAN) code is available from the author.

The responsibility for the wording and views expressed in this descriptive summary lies with MRC, and not with the author of this report.

List of Contents

A BIVARIATE C^2 CLOUGH-TOCHER SCHEME

Peter Alfeld*

1. Introduction

In order to save space, the reader is assumed to be familiar with the introduction and section 2. of Alfeld, 1984. There, a bivariate and a trivariate C^1 Clough-Tocher scheme were derived. Having obtained a C^1 scheme, it is natural to use the same techniques to construct a (bivariate) C^2 scheme. Such a scheme would be useful e.g. for the design of aircrafts, where C^2 smoothness of surfaces is required. Interpolation by a function that is polynomial on each macrotriangle would require a polynomial of degree at least nine and data through fourth order (Zenisek, 1970). In this paper, we derive a scheme that is piecewise quintic on each macrotriangle and that requires only C^2 data.

Farin, 1980, has shown that a piecewise polynomial C^2 scheme cannot be constructed on the simple Clough-Tocher split. The difficulty can be traced to the fact that each vertex angle is divided into only two parts. For a C^2 scheme, a division into at least three subangles is required. The work described in this paper grew out of efforts by Arner, Barnhill, Farin, and Little to construct a C^2 scheme for the split depicted in figure 1. That split generates the minimum number of triangles (seven) while trisecting each angle at the vertices. It is still an unsettled question if a C^2 scheme exists for the minimal split.

*Department of Mathematics, University of Utah, Salt Lake City, Utah 84112

Sponsored by the United States Army under Contract No. DAAG29-80-C-0041. This research was also supported by the Department of Energy under Contract No. DE-AC02-82ER12046 to the University of Utah and partly by a faculty grant from the University of Utah Research Committee.

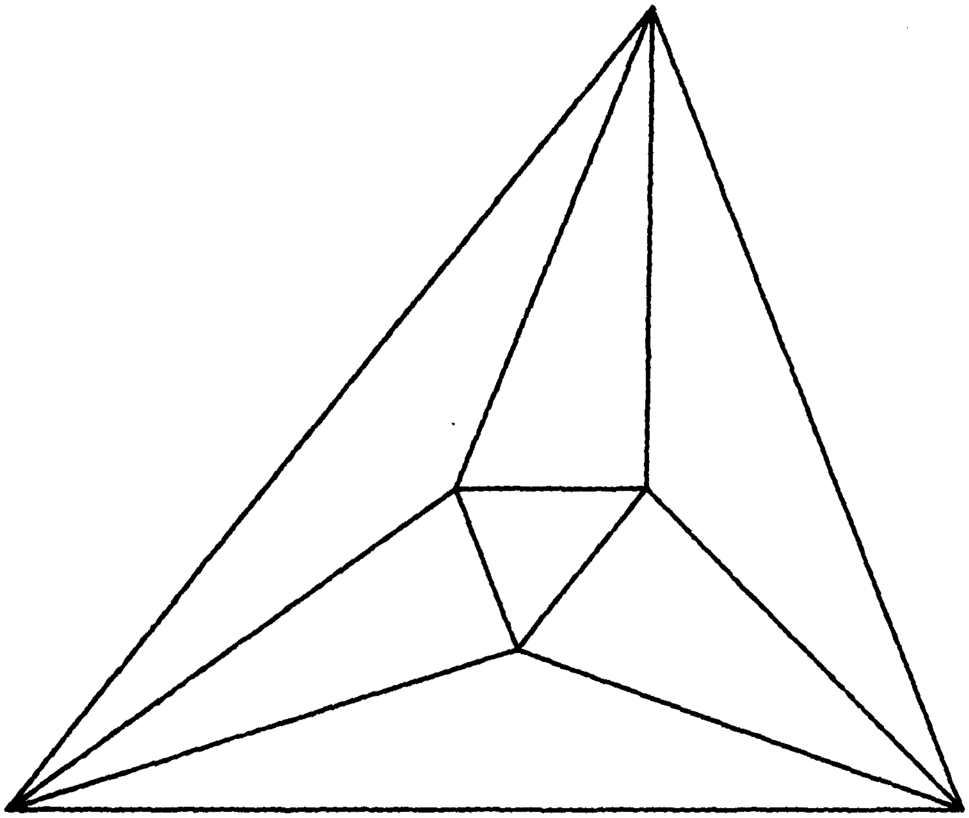


Figure 1: A Minimal Split

In the approach described here, the Clough-Tocher split is applied twice. First we divide the macrotriangle into three *subtriangles*, and then we divide each subtriangle into three *microtriangles*. Thus each macrotriangle is the union of nine microtriangles. We refer to the centroid of the macrotriangle simply as *the centroid*, and to the centroids of the subtriangles as the *subcentroids*. The construction is illustrated in figure 2. We assume we are given C^2 data (i.e. values of position, gradient and Hessian) at the vertices, and construct an interpolant that is quintic on each microtriangle and twice differentiable everywhere on the triangulated domain.

The major difference between the approach in Alfeld, 1984, and that described here is that for the C^1 schemes it was possible to identify in a natural way one Bezier ordinate with each of the conditions defining the interpolant. This turns out to be impossible in the C^2 case.

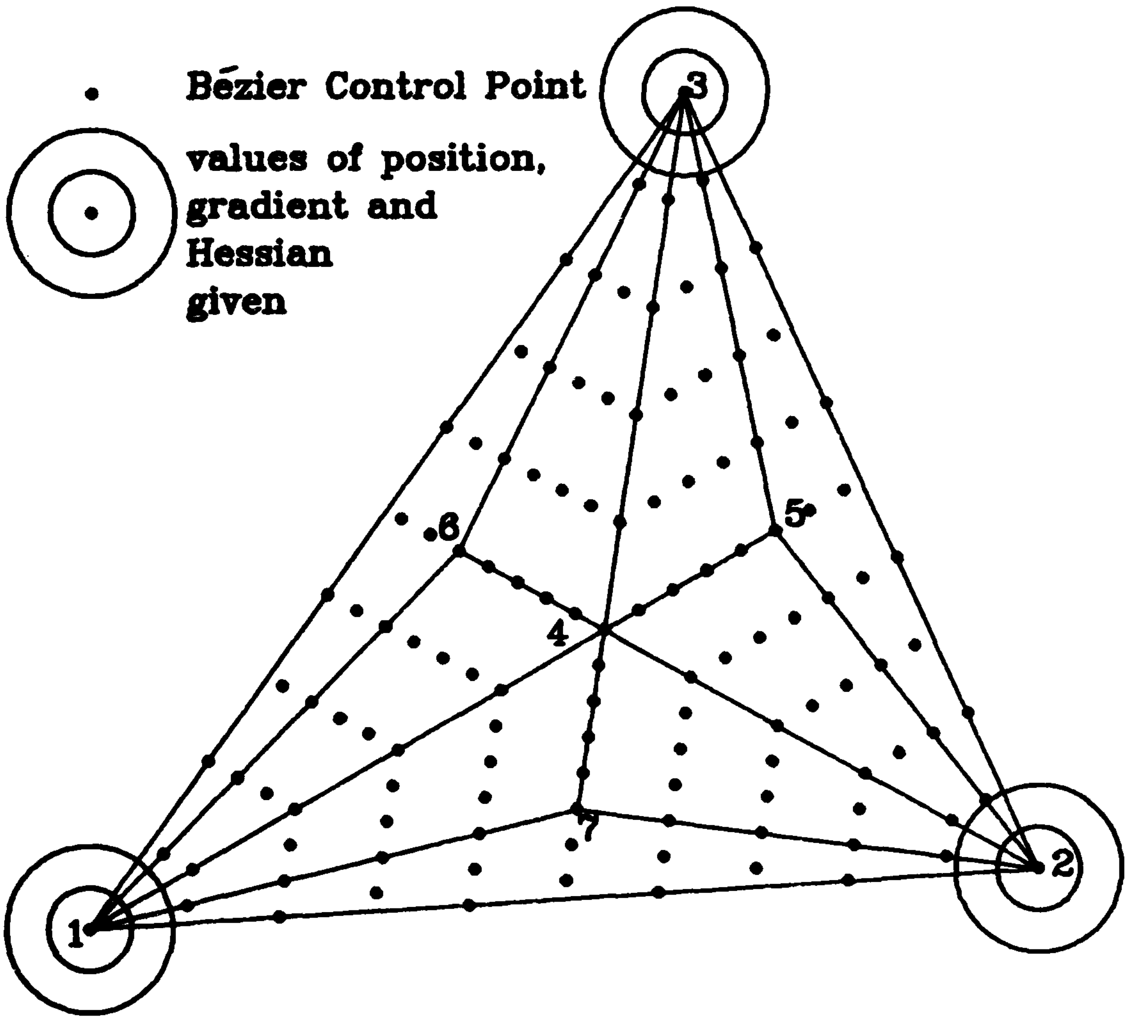


Figure 2: Data Stencil, the Double Clough–Tocher Split and the Piecewise Quintic Bézier Net

2.0 Notation and Internal Continuity

We consider a general macrotriangle with vertices V_1, V_2, and V_3, and denote the centroid by $V_4 = (V_1 + V_2 + V_3)/3$. The subcentroids are denoted by $V_5 = (V_2 + V_3 + V_4)/3$, $V_6 = (V_1 + V_3 + V_4)/3$, and $V_7 = (V_1 + V_2 + V_4)/3$. Figure 2 illustrates the construction.

It is significant that the triples V_1, V_4, V_5, V_2, V_4, V_6, and V_3, V_4, V_7 are each colinear. This can be seen for example by writing

$$\begin{aligned} V_5 &= (V_4 + V_2 + V_3)/3 \\ &= (V_4 + (3V_4 - V_1 - V_3) + (3V_4 - V_1 - V_2))/3 \\ &= (7V_4 - (V_1 + V_2 + V_3) - V_1)/3 \\ &= (4V_4 - V_1)/3. \end{aligned}$$

The technique of expressing the location of a point in terms of different sets of internal and external vertices will be used frequently in the sequel. To obtain smoothness conditions we will express cross-boundary directions in terms of the vertices of each of the adjacent microtriangles, thereby facilitating differentiation on each of the microtriangles. This approach is described in detail in Alfeld, 1984, and is further exemplified in section 2.2.2 below.

The location of a general point P is expressed as

$$P = \sum_{i=1}^{7} b_i V_i$$

where the b_i are the piecewise linear cardinal functions defined on the triangulation of the macrotriangle by $b_i(V_j) = \delta_{ij}$, δ being the Kronecker Delta. The interpolant to be constructed is of the form

$$p(P) = \sum_{\sum_{j=1}^{7} i_j = 5} \frac{5!}{\prod_{j=1}^{7} (i_j!)} c_{i_1 i_2 \ldots i_7} \prod_{j=1}^{7} b_j^{i_j}$$

where, by convention, $0^0 := 1$.

The function p is continuous on the macrotriangle and contains 121 free parameters. (Remarkably, although to all appearances coincidentally, this is the same number of parameters as is available for the construction of the trivariate C^1 Clough-Tocher scheme, see Alfeld, 1984.) For a given point P, the generalized barycentric coordinates can be easily computed as follows:

1. Compute the barycentric coordinates with respect to the macrotriangle, i.e. write $P = B_1V_1 + B_2V_2 + B_3V_3$. Let $\{i,j,k\} = \{1,2,3\}$ and suppose $B_i < B_j$ and $B_i < B_k$. Then P lies in the subtriangle with vertices V_j, V_k, and V_4. The barycentric coordinates with respect to that triangle, a_j, a_k, and a_4, say, are given by $a_4 = 3B_i$, $a_j = B_j - B_i$, $a_k = B_k - B_i$.

2. After relabeling, repeat step 1 on the subtriangle to locate the microtriangle containing P.

The microtriangle with vertices V_i, V_j, and V_k will be denoted by T_{ijk}.

2.1 Interpolation to Vertex Data

Since it is impossible to associate in a natural and unique way Bezier ordinates with conditions on the interpolant we start by considering interpolation to the vertex data. This will keep a larger part of the linear system triangular. Thus we do not assume that we have already enforced smoothness. On the contrary, interpolation at the vertices will force C^2 smoothness at the vertices. It affects all Bezier ordinates that are no more

than two levels removed from the vertices. A simple count using figure 2 or 3 shows that 45 conditions have to be satisfied. These are given as equations 1-45 in the appendix. The technique for the derivation is analogous to that in Alfeld, 1984, and in Barnhill and Farin, 1981. Note that the data are expressed as derivatives in the direction of edges. They have to be supplied consistently. This is automatic if all such data are computed f· m cartesian derivatives through second order.

2.2 Internal First Order Differentiability

To force differentiability of the interpolant everywhere in the macrotriangle, we proceed in stages, distinguishing between differentiability in the interior of subtriangles, differentiability across lines separating subtriangles, and differentiability at the centroid. The complete set of conditions is described in figure 3. Each stencil gives coefficients of a linear combination of the Bezier ordinates that must be zero. For each type, one representative stencil is depicted. For better legibility, it has been moved out of the drawing of the macrotriangle. All other stencils of the same type can be obtained by drawing them about the appropriately marked centers. Stencils in subsequent figures should be interpreted similarly.

2.2.1 C^1 Conditions in the Interior of Subtriangles

As was shown in Alfeld, 1984, the relevant condition is that the Bezier ordinates in the interior of internal edges equal the averages of the three neighboring vertices. However, interpolation at the vertices already imposed two conditions on each internal edge emanating from a vertex. Consider the difference between the two cross-boundary derivatives derived from two adjacent microtriangles, expressed as a univariate Bezier polynomial.

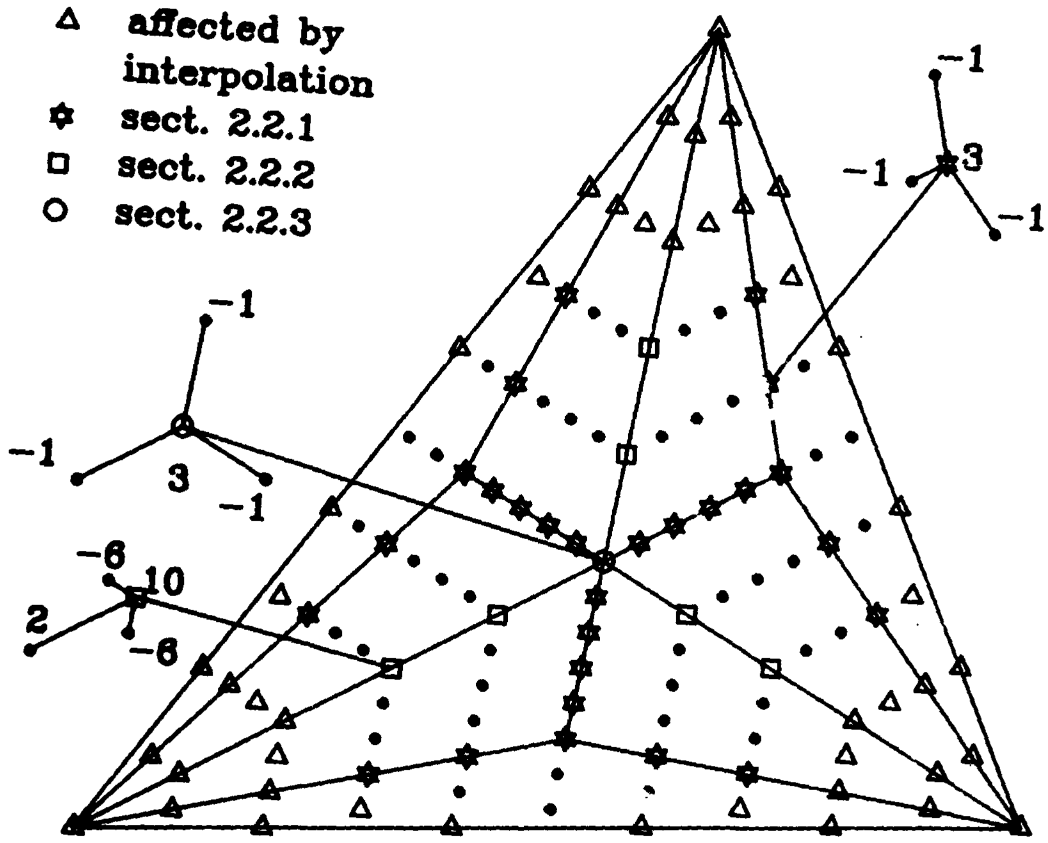


Figure 3: First Order Differentiability Stencils and their Centers

Interpolation at the vertices forces the value of that difference as well as its tangential derivative to vanish at the vertex. Thus the two Bezier ordinates closest to the vertex will be zero, so that only the Bezier ordinates shown in figure 3 are affected. The total number of conditions so obtained is 27.

The specific stencil illustrated in figure 3 means that algebraically

$$3c_{0010400} - c_{0110300} - c_{0020300} - c_{0011300} = 0$$

2.2.2. C^1 Conditions on Internal Edges from Vertices to the Centroid

We proceed similarly as in the analysis leading to the equations in the preceding subsection. Consider for example e_{14}, which is shared by the microtriangles T_{146} and T_{147}. In order to obtain a symmetric stencil we consider the derivative in the direction $e_{67} = V_7 - V_6$ which we express on T_{146} as $e_{67} = (5e_{64} + e_{61})/3$ and on T_{147} as $e_{67} = (e_{17} + 5e_{47})/3$. Then we differentiate the interpolant on each microtriangle, restrict to e_{14}, and consider the difference a univariate Bezier polynomial in b_4 and b_7. The two Bezier ordinates of that polynomial that are closest to the vertex vanish because of interpolation to the vertex data, and the one at the centroid will be dealt with in the next subsection. So we require that the remaining two vanish, yielding e.g. the condition

$$10c_{2003000} + 2c_{300200} - 6c_{2002010} - 6c_{2002001} = 0$$

This is the particular equation depicted in figure 3. On the other edges we proceed analogously. Altogether we obtain six conditions.

2.2.3 C^1 Condition at the Centroid

In Alfeld, 1984, it was shown that for the single Clough-Tocher split the appropriate condition on the Bezier ordinate at the centroid is that it equal

the average of its three neighbors. That argument does not apply in the present case because the interpolant is piecewise polynomial on each of the subtriangles, rather than polynomial. However, the result is the same!

The following geometric argument (which superseded an elaborate algebraic one) is due to C.S. Petersen (private communication): For the interpolant to be C^2 at V_4, the set S, say, of all Bezier control points corresponding to $c_{xxxixxx}$ where $i > 3$ and the x's are arbitrary must lie in the same plane. The previously imposed conditions imply that any arrow shaped subset of S pointing to the centroid (like $\{c_{0005000}, c_{0104000}, c_{0014000}, c_{0004100}\}$ or $\{c_{0005000}, c_{0104000}, c_{0004100}, c_{0004001}\}$) is coplanar. Obviously, all points in S will be coplanar if the set corresponding to $\{c_{1004000}, c_{0104000}, c_{0014000}, c_{0005000}\}$ (which links all those arrows) is coplanar. This yields the condition

$$3c_{0005000} - c_{1004000} - c_{0104000} - c_{0014000} = 0$$

as depicted in figure 3.

The total number of first order differentiability conditions is 33.

2.3 Internal Second Order Differentiability

When forcing continuity of second derivatives, care is needed that the linear system describing the interpolant does not become overdetermined. The mechanics of deriving conditions are as in the C^1 case. We proceed in several stages.

2.3.1 C^2 Conditions at Vertices

Second order differentiability is enforced by interpolation to the second order vertex data.

2.3.2 C^2 Conditions at Subcentroids

Second order differentiability is serendipitously implied by the first order conditions, for details see Alfeld, 1984.

2.3.3 C^2 Conditions on Lines from Vertices to Subcentroids

Consider the difference between suitable second order cross-boundary derivatives as they have been obtained from two neighboring microtriangles. That difference is a univariate cubic polynomial with four degrees of freedom. We must enforce conditions that make the difference zero. One condition each has been imposed at the vertex and at the subcentroid, leaving two degrees of freedom.

For example, on edge e_{27} we consider the second order derivative in the direction of e_{14} and obtain

$$c_{3001001} - c_{3100001} - c_{2200001} + 3c_{2100002} + c_{2002001} - 3c_{2001002} = 0$$

and one more such condition that can be obtained by shifting the stencil of the above equation towards V_7. Altogether there are 12 conditions of this type.

2.3.4 C^2 Conditions on Lines from Subcentroids to the Centroid

Since we have not yet enforced a second order condition at the centroid, one might expect that three conditions are needed. However, the conditions described in section 2.3.3 imply that all first order tangential derivatives of second order cross boundary derivatives are continuous at the subcentroid.

To see this consider e.g. point V_6 and edge e_{46}. It is sufficient to consider any particular second order cross-boundary derivative. For convenience we choose the derivative

$$\frac{\partial^2}{\partial e_{16}\partial e_{36}}$$

Thus we are interested in the continuity of $\frac{\partial^3}{\partial e_{46}\partial e_{16}\partial e_{36}}$ at V_6.

That derivative agrees between T_{346} and T_{136} since it is a tangential derivative of a second order cross-boundary derivative in the direction of e_{36} and because we have already forced second order differentiability across e_{36}. Similarly it agrees between T_{146} and T_{147}. Hence it agrees between T_{346} and T_{146} across e_{46} which is what we wanted to show.

Because of this serendipitous extra piece of smoothness we have to enforce only two conditions on each line from a subcentroid to the centroid. Considering the second order derivative in the direction of e_{13} across e_{64} we obtain the symmetric condition

$$-c_{0013010} - c_{0022010} + 3c_{0012020} - 3c_{1002020} + c_{2002010} + c_{1003010} = 0$$

This particular condition is depicted in figure 4, and there are six such conditions altogether.

Note that the conditions derived in this subsection are analogous to those in the previous subsection, which is as one would expect because of the symmetry in the subtriangles. However, they required a separate derivation because no interpolation takes place at the centroid.

2.3.5 C^2 Conditions at the Centroid

This follows serendipitously as at the subcentroids.

2.3.6 C^2 Conditions on Lines from Vertices to the Centroid

It appears at first that as in section 2.3.3 two conditions have to be imposed on each line, yielding six conditions altogether. However, one of them is implied by the other five.

Consider, for example, the second order e_{57} derivative across edge e_{24}. On T_{247} we can write $e_{57} = (e_{27} + 5e_{47})/3$ and on T_{245} we have $e_{57} = (e_{52} + 5e_{54})/3$. Carrying out the manipulations yields the condition

$$c_{0202100} - c_{0202001} + 5c_{0103100} - 5c_{0103001} - 3c_{0102200} + 3c_{0102002} = 0$$

which is depicted in figure 4. The number of conditions so obtained equals six.

Now consider the condition with c_{001400} at its center. Making the corresponding Bezier ordinate of the difference between two second order derivatives zero is equivalent to making the tangential derivative at the centroid zero. So consider for example the derivative

$$D = \frac{\partial^3}{\partial e_{14} \partial e_{24} \partial e_{34}} (V_4)$$

and assume that we have enforced the other five C^2 conditions. It can be seen by a refinement of the argument in section 2.3.4 that D is indeed continuous. The key observation is that the relevant triples of vertices, subcentroids, and centroid are colinear, and that hence D is a tangential derivative of a second order cross-boundary derivative in each microtriangle sharing the centroid.

Thus in this subsection we obtain five conditions altogether.

2.4 Intertriangular Smoothness

Here we must consider perpendicular cross-boundary derivatives. The first order such derivative is in general quartic with five parameters, but we are only given four data. So we require additionally that the leading coefficient vanishes, making the derivative cubic. Similarly, we require that the second order perpendicular cross-boundary derivative be linear.

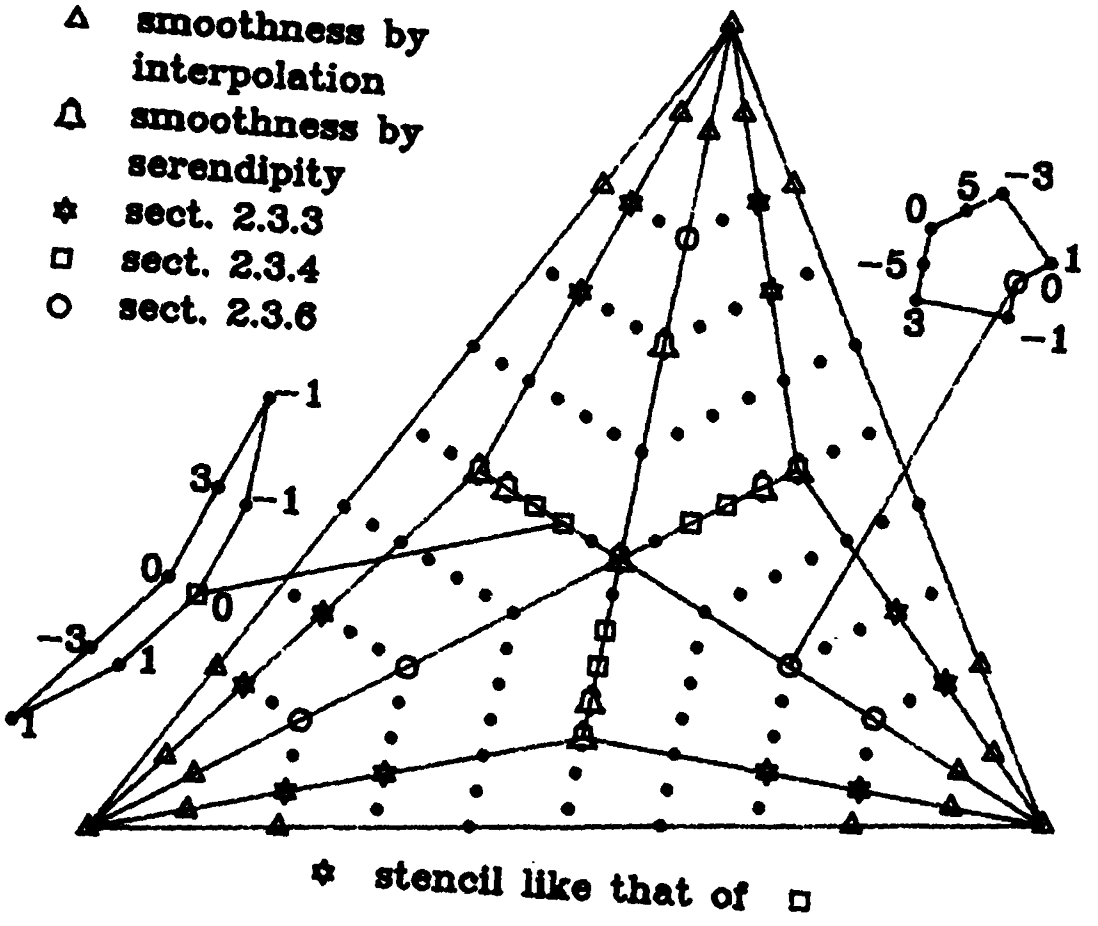


Figure 4: Second Order Differentiability Stencils and their Centers

It is not necessary to consider perpendicular cross-boundary directions of unit length. Instead, the normalization can be chosen so as to be appropriate to the geometry of the macrotriangle. For example, on edge e_{12} we define the normal n_{12} by

$$n_{12} = e_{17} + \gamma_3 e_{12} \quad \text{where} \quad \gamma_3 = -(e_{17}^T e_{12})/(e_{17}^T e_{12})$$

and express the cross-boundary derivatives as univariate polynomials in b_1. Setting the leading coefficient of the first order cross-boundary derivative equal to zero yields the stencil given at the bottom of figure 5. Similarly, setting the two leading coefficients of the second order cross-boundary derivative to zero yields two similar conditions which are depicted in figure 5 on other edges. All twelve such conditions can be obtained from figure 5 by rotating the labeling and modifying appropriately the subscripts of γ.

The total number of conditions for e tratriangular smoothness is twelve.

2.5 Condensation of Parameters

At this stage, 111 conditions have been imposed, leaving 10 degrees of freedom. In disposing of the ambiguity one would like to meet three objectives:

1. The maximum possible degree of precision (cubic) is maintained.

2. The interpolant is independent of the labeling of the vertices.

3. The additional conditions are simple. For example, one would not like to require interpolation to additional data that would have to be made up.

All of these objectives are met by the following requirements:

1. The interpolant is quartic on lines from vertices to the centroid (3 conditions). This requirement is formulated algebraically by setting the fifth order tangential derivatives (which are constants) equal to zero.

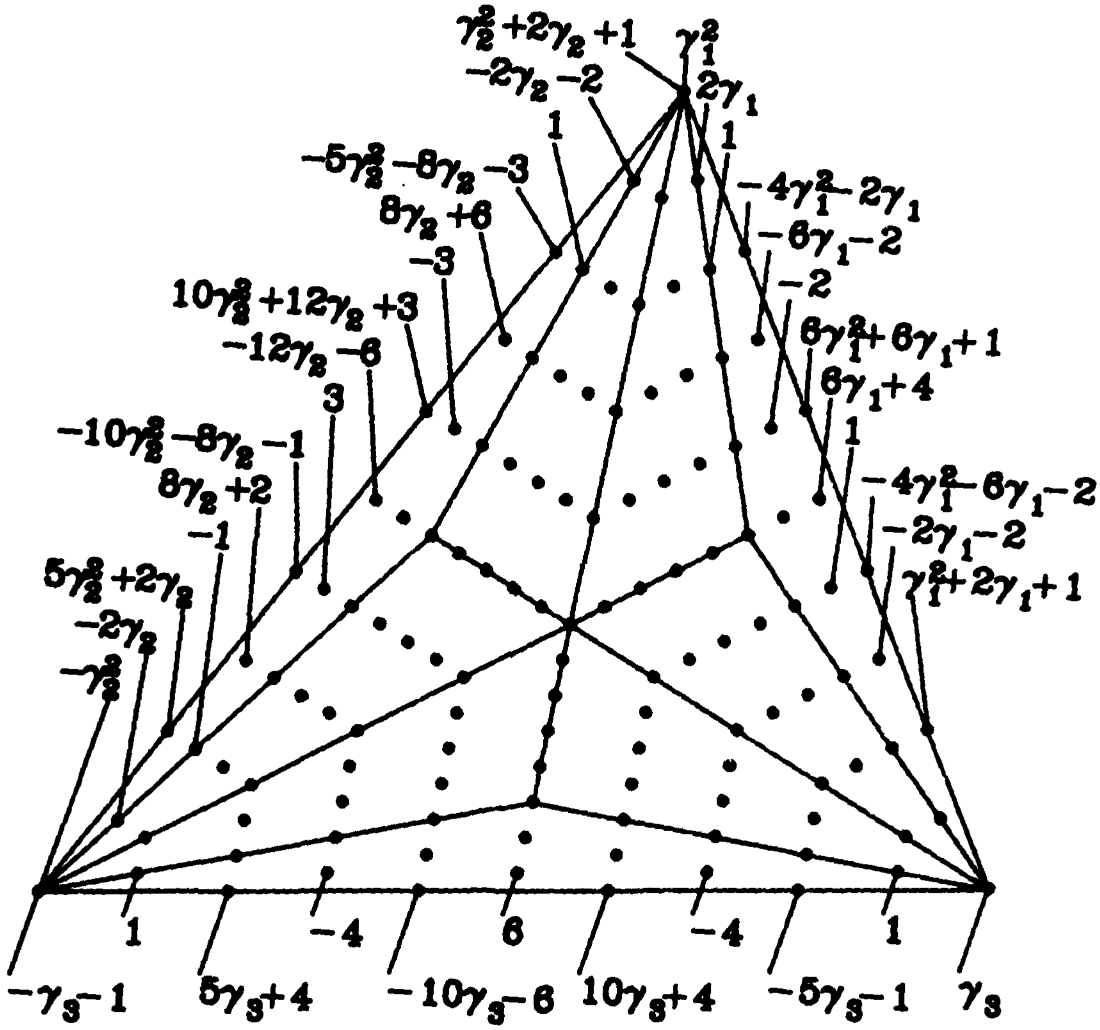


Figure 5: Intertriangular Smoothness

2. The interpolant is cubic on lines from subcentroids to the centroid (6 conditions). Here we set fourth and fifth derivatives to zero at the centroid.

3. The sum of tangential fourth order derivatives on the lines from the vertices to the centroid equals zero. More precisely

$$\frac{\partial^4 p}{\partial e_{14}^4} + \frac{\partial^4 p}{\partial e_{24}^4} + \frac{\partial^4 p}{\partial e_{34}^4} = 0$$

These requirements give rise to particularly simple stencils that are depicted in figure 6.

2.6 The Solution of the Linear System

The analysis in the previous subsections defines a linear system of 121 equations for the 121 coefficients of the interpolant. That system was set up and solved using the symbol manipulation language REDUCE (Hearn, 1983). The first 48 equations (corresponding to the interpolation conditions and the requirement that first order perpendicular cross-boundary derivatives be quartic) are lower triangular. The remaining equations were reduced to lower triangular form, and a listing of the coefficients in Forward Elimination form is given in the appendix.

2.7 Precision of the Interpolant.

The interpolant developed here reproduces all bivariate cubic polynomials exactly. This follows as in section 3.6 of Alfeld, 1984.

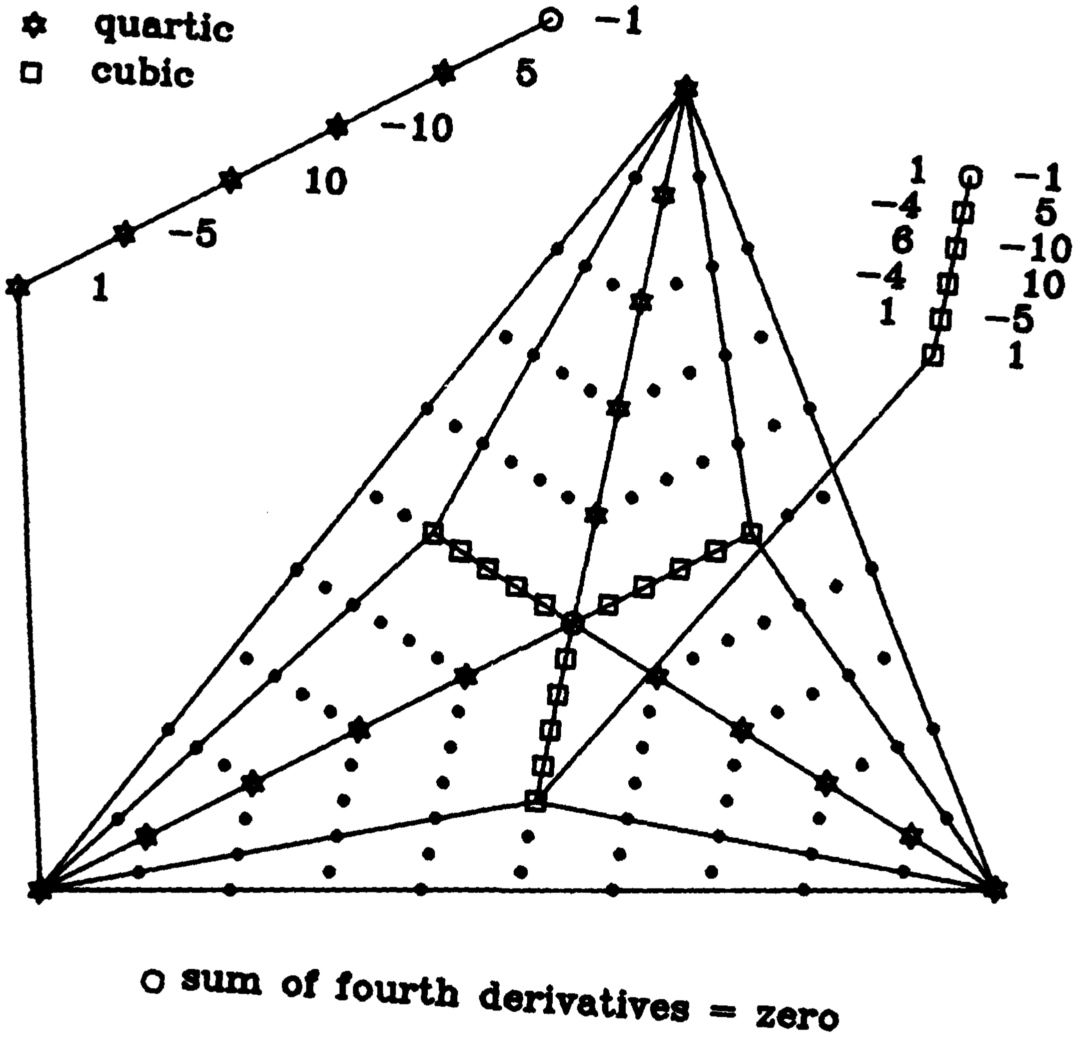


Figure 6: Condensation of Parameters

3. Computational Aspects.

It is impractical to base an implementation of the interpolant on the printed formulas given in the appendix. Instead, any code should be written directly by the symbol manipulation language used to solve the linear system. A pilot version of such a code is available from the author. It code has been tested by the analyzing tool **MICROSCOPE** (Alfeld and Harris, 1984) and it was verified that the code does indeed posses the smoothness, interpolation, and precision properties that are implied by the mathematical construction.

Conclusions

The scheme developed here is the first explicitly given piecewise polynomial C^2 interpolant for triangular C^2 data.

Acknowledgements

The author has benefitted from the stimulating environments provided by the Computer Aided Geometric Design Group at the University of Utah and by the Mathematics Research Center at the University of Wisconsin. The figures were generated using the software package **PLOT79** (Beebe, 1980).

REFERENCES

P. Alfeld (1984), A Trivariate Clough-Tocher Scheme for Tetrahedral Data, submitted for publication.

P. Alfeld and B. Harris (1984), MICROSCOPE: A Software Tool for the Analysis of Multivariate Functions, in preparation.

R.E. Barnhill and G. Farin (1981), C^1 Quintic Interpolation over Triangles: Two Explicit Representations, Int. J. for Num. Meth. in Eng. 17, pp. 1763-1778

N.H.F. Beebe, (1980), A User's Guide to PLOT79, Departments of Physics and Chemistry, University of Utah, Salt Lake City, UT 84112

G. Farin (1980), Bezier Polynomials over Triangles and the Construction of Piecewise C^r Polynomials, Report TR/91, Depart. of Math., Brunel University, Uxbridge, Englan

A. C. Hearn (1983), REDUCE User's Manual, Version 3.0,Rand Publication CP78(4/83), The Rand Corporation, Santa Monica, CA 90406

A. Zenisek (1970), Interpolation polynomials on the triangle, Numer. Math. 15, pp. 283-296

Appendix: Coefficients of Interpolant

Following are expressions for the coefficients of the interpolant as the have been generated by **REDUCE**. To eliminate typing and type-setting errors the results have been reproduced photographically. The organization of the listing is like that in Alfeld, 1984. The parameters γ_i have been abbreviated by Gi.

```
1:  c5000000 = F(0,V1)

2:  c4100000 = (5*c5000000 + F(12,V1))/5

3:  c3200000 = ( - 20*c5000000 + 40*c4100000 + F(1212,V1))/20

4:  c4000001 = (5*c5000000 + F(17,V1))/5

5:  c3000002 = ( - 20*c5000000 + 40*c4000001 + F(1717,V1))/20

6:  c3100001 = ( - 20*c5000000 + 20*c4100000 + 20*c4000001 + F(1217,V1))/20

7:  c4010000 = (5*c5000000 + F(13,V1))/5

8:  c3020000 = ( - 20*c5000000 + 40*c4010000 + F(1313,V1))/20

9:  c4000010 = (5*c5000000 + F(16,V1))/5

10: c3000020 = ( - 20*c5000000 + 40*c4000010 + F(1616,V1))/20

11: c3010010 = ( - 20*c5000000 + 20*c4010000 + 20*c4000010 + F(1316,V1))/20

12: c4001000 = (5*c5000000 + F(14,V1))/5

13: c3002000 = ( - 20*c5000000 + 40*c4001000 + F(1414,V1))/20

14: c3001010 = ( - 20*c5000000 + 20*c4001000 + 20*c4000010 + F(1416,V1))/20

15: c3001001 = ( - 20*c5000000 + 20*c4001000 + 20*c4000001 + F(1417,V1))/20

16: c0500000 = F(0,V2)

17: c0410000 = (5*c0500000 + F(23,V2))/5

18: c0320000 = ( - 20*c0500000 + 40*c0410000 + F(2323,V2))/20

19: c0400100 = (5*c0500000 + F(25,V2))/5

20: c0300200 = ( - 20*c0500000 + 40*c0400100 + F(2525,V2))/20

21: c0310100 = ( - 20*c0500000 + 20*c0410000 + 20*c0400100 + F(2325,V2))/20

22: c1400000 = (5*c0500000 + F(21,V2))/5

23: c2300000 = (40*c1400000 - 20*c0500000 + F(2121,V2))/20

24: c0400001 = (5*c0500000 + F(27,V2))/5

25: c0300002 = ( - 20*c0500000 + 40*c0400001 + F(2727,V2))/20

26: c1300001 = (20*c1400000 - 20*c0500000 + 20*c0400001 + F(2127,V2))/20

27: c0401000 = (5*c0500000 + F(24,V2))/5
```

```
28:  c0302000 = ( - 20*c0500000 + 40*c0401000 + F(2424,V2))/20

29:  c0301001 = ( - 20*c0500000 + 20*c0401000 + 20*c0400001 + F(2427,V2))/20

30:  c0301100 = ( - 20*c0500000 + 20*c0401000 + 20*c0400100 + F(2425,V2))/20

31:  c0050000 = F(0,V3)

32:  c1040000 = (5*c0050000 + F(31,V3))/5

33:  c2030000 = (40*c1040000 - 20*c0050000 + F(3131,V3))/20

34:  c0040010 = (5*c0050000 + F(36,V3))/5

35:  c0030020 = ( - 20*c0050000 + 40*c0040010 + F(3636,V3))/20

36:  c1030010 = (20*c1040000 - 20*c0050000 + 20*c0040010 + F(3136,V3))/20

37:  c0140000 = (5*c0050000 + F(32,V3))/5

38:  c0230000 = (40*c0140000 - 20*c0050000 + F(3232,V3))/20

39:  c0040100 = (5*c0050000 + F(35,V3))/5

40:  c0030200 = ( - 20*c0050000 + 40*c0040100 + F(3535,V3))/20

41:  c0130100 = (20*c0140000 - 20*c0050000 + 20*c0040100 + F(3235,V3))/20

42:  c0041000 = (5*c0050000 + F(34,V3))/5

43:  c0032000 = ( - 20*c0050000 + 40*c0041000 + F(3434,V3))/20

44:  c0031100 = ( - 20*c0050000 + 20*c0041000 + 20*c0040100 + F(3435,V3))/20

45:  c0031010 = ( - 20*c0050000 + 20*c0041000 + 20*c0040010 + F(3436,V3))/20

  46:  c2200001 = (c5000000*G3 + c5000000 - 5*c4100000*G3 - 4*c4100000 -
c4000001 + 10*c3200000*G3 + 6*c3200000 + 4*c3100001 - 10*c2300000* G3  -
4*c2300000 + 5*c1400000*G3 + c1400000 + 4*c1300001 - c0500000* G3 - c0400001)
/6

  47:  c0220100 = (c0500000*G1 + c0500000 - 5*c0410000*G1 - 4*c0410000 -
c0400100 + 10*c0320000*G1 + 6*c0320000 + 4*c0310100 - 10*c0230000* G1  -
4*c0230000 + 5*c0140000*G1 + c0140000 + 4*c0130100 - c0050000* G1 - c0040100)
/6

  48:  c2020010 = ( - c5000000*G2 + 5*c4010000*G2 + c4010000 - c4000010 -
10*c3020000*G2 - 4*c3020000 + 4*c3010010 + 10*c2030000*G2 + 6* c2030000 -
5*c1040000*G2 - 4*c1040000 + 4*c1030010 + c0050000*G2 + c0050000 - c0040010)/6
```

49: c0201002 = (14661*c5000000*G3**2 + 29322*c5000000*G3 + 5535*
c5000000*G2**2 + 16315*c5000000 - 22707*c4100000*G3**2 - 16092* c4100000*G3 +
6615*c4100000 - 16605*c4010000*G2**2 - 11070*c4010000* G2 - 5190*c4001000 +
11070*c4000010*G2 - 29322*c4000001*G3 - 29322* c4000001 - 55782*c3200000*G3**2
- 127656*c3200000*G3 - 57213*c3200000 + 16092*c3100001*G3 - 10665*c3100001 +
11070*c3020000*G2**2 + 22140* c3020000*G2 + 5535*c3020000 - 22140*c3010010*G2 -
13635*c3010010 + 5360*c3002000 + 4815*c3000020 + 15021*c3000002 +
156978*c2300000* G3**2 + 186300*c2300000*G3 + 35937*c2300000 +
127656*c2200001*G3 + 98280*c2200001 + 11070*c2030000*G2**2 - 5535*c2030000 +
5940*c2020010 - 129087*c1400000*G3**2 - 71874*c1400000*G3 - 186300*c1300001*G3
- 90585*c1300001 - 16605*c1040000*G2**2 - 22140*c1040000*G2 - 5535* c1040000 +
22140*c1030010*G2 + 8505*c1030010 + 35937*c0500000*G3**2 - 5643*c0500000*G1**2
- 11286*c0500000*G1 - 2021*c0500000 + 16443* c0410000*G1**2 + 21600*c0410000*G1
+ 5157*c0410000 - 15030*c0401000 + 11286*c0400100*G1 + 11286*c0400100 +
71874*c0400001*G3 - 9342* c0320000*G1**2 + 2916*c0320000*G1 + 6615*c0320000 -
21600*c0310100* G1 - 8235*c0310100 + 24392*c0302000 + 16632*c0301001 -
5283*c0300200 + 36297*c0300002 - 14202*c0230000*G1**2 - 25488*c0230000*G1 -
6129* c0230000 - 2916*c0220100*G1 - 8586*c0220100 + 18873*c0140000*G1**2 +
12258*c0140000*G1 + 25488*c0130100*G1 + 14823*c0130100 - 6129* c0050000*G1**2 +
5535*c0050000*G2**2 + 11070*c0050000*G2 + 7189* c0050000 - 5190*c0041000 -
12258*c0040100*G1 - 11070*c0040010*G2 - 11070*c0040010 + 5360*c0032000 -
5769*c0030200 + 4815*c0030020)/49896

50: c0004100 = (- 2681775*c5000000*G3**2 - 5363550*c5000000*G3 -
810945*c5000000*G2**2 - 1594689*c5000000 + 4242915*c4100000*G3**2 +
3122280*c4100000*G3 - 1120635*c4100000 + 2992005*c4010000*G2**2 +
1621890*c4010000*G2 - 3548930*c4001000 - 1621890*c4000010*G2 +
5363550*c4000001*G3 + 5363550*c4000001 + 9846090*c3200000*G3**2 +
22814460*c3200000*G3 + 10286595*c3200000 - 3122280*c3100001*G3 +
1715445*c3100001 - 3858570*c3020000*G2**2 - 4362120*c3020000*G2 -
810945*c3020000 + 4362120*c3010010*G2 + 2147715*c3010010 + 3588200* c3002000 -
741825*c3000020 - 2834055*c3000002 - 28178010*c2300000*G3**2 -
33541560*c2300000*G3 - 6484185*c2300000 - 22814460*c2200001*G3 -
17822430*c2200001 + 1733130*c2030000*G2**2 + 3355020*c2030000*G2 +
1370115*c2030000 - 3355020*c2020010*G2 - 2247750*c2020010 + 23254965*
c1400000*G3**2 + 12968370*c1400000*G3 + 33541560*c1300001*G3 +
16244955*c1300001 + 196155*c1040000*G2**2 - 111240*c1040000*G2 -
307395*c1040000 + 111240*c1030010*G2 + 581445*c1030010 - 6484185*
c0500000*G3**2 - 49815*c0500000*G1**2 - 99630*c0500000*G1 + 760707* c0500000 +
249075*c0410000*G1**2 + 398520*c0410000*G1 + 149445* c0410000 -
2166110*c0401000 + 99630*c0400100*G1 + 99630*c0400100 - 12968370*c0400001*G3 -
498150*c0320000*G1**2 - 597780*c0320000*G1 - 149445*c0320000 -
398520*c0310100*G1 - 165915*c0310100 + 209840* c0302000 - 3409560*c0301001 +
33345*c0300200 - 6636465*c0300002 + 498150 *c0230000*G1**2 + 398520*c0230000*G1
+ 49815*c0230000 + 597780* c0220100*G1 + 465210*c0220100 + 10228680*c0201002 -
249075*c0140000* G1**2 - 99630*c0140000*G1 - 398520*c0130100*G1 -
66285*c0130100 + 49815*c0050000*G1**2 - 251775*c0050000*G2**2 -
503550*c0050000*G2 + 962187*c0050000 - 4183310*c0041000 + 99630*c0040100*G1 +
503550* c0040010*G2 + 503550*c0040010 + 4111400*c0032000 + 132975*c0030200 -
182655*c0030020)/4532220

```
51:  c0004010 = (707670*c5000000*G3**2 + 1415340*c5000000*G3 + 154170
*c5000000*G2**2 + 539058*c5000000 - 1197045*c4100000*G3**2 - 978750*
c4100000*G3 + 218295*c4100000 - 588195*c4010000*G2**2 - 308340* c4010000*G2 +
543610*c4001000 + 308340*c4000010*G2 - 1415340*c4000001 *G3 - 1415340*c4000001
- 2288520*c3200000*G3**2 - 5555790*c3200000* G3 - 2559600*c3200000 +
978750*c3100001*G3 - 331425*c3100001 + 811080* c3020000*G2**2 +
868050*c3020000*G2 + 154170*c3020000 - 868050* c3010010*G2 - 413505*c3010010 -
553300*c3002000 + 142290*c3000020 + 740070*c3000002 + 6971130*c2300000*G3**2 +
8386470*c2300000*G3 + 1633635*c2300000 + 5555790*c2200001*G3 + 4403565*c2200001
- 445770* c2030000*G2**2 - 754110*c2030000*G2 - 279855*c2030000 + 754110*
c2020010*G2 + 475065*c2020010 - 5826870*c1400000*G3**2 - 3267270* c1400000*G3 -
8386470*c1300001*G3 - 4088070*c1300001 + 40230*c1040000 *G2**2 +
137430*c1040000*G2 + 97200*c1040000 - 137430*c1030010*G2 - 173880*c1030010 +
1633635*c0500000*G3**2 - 103158*c0500000 + 216340* c0401000 +
3267270*c0400001*G3 - 20520*c0310100 + 268820*c0302000 + 841320*c0301001 -
20520*c0300200 + 1666035*c0300002 - 41040*c0220100 - 2523960*c0201002 -
20520*c0130100 + 28485*c0050000*G2**2 + 56970* c0050000*G2 - 168645*c0050000 +
686200*c0041000 - 56970*c0040010*G2 - 56970*c0040010 - 670900*c0032000  -
20520*c0030200 + 16605*c0030020 + 1018710*c0004100)/199260
```

```
52:  c0011300 = ( - 67049100*c5000000*G3**2 - 134098200*c5000000*G3 +
155878560*c5000000*G2**2 - 1110940473*c5000000 + 335245500*c4100000* G3**2  +
536392800*c4100000*G3 + 201147300*c4100000 - 497174220* c4010000*G2**2 -
311757120*c4010000*G2 + 3452811905*c4001000 + 311757120*c4000010*G2 +
134098200*c4000001*G3 + 134098200*c4000001 - 670491000*c3200000*G3**2 -
804589200*c3200000*G3 - 201147300*c3200000 - 536392800*c3100001*G3 -
320686560*c3100001 + 429911280*c3020000* G2**2 + 682591320*c3020000*G2 +
155878560*c3020000 - 682591320*c3010010 *G2 - 393365160*c3010010 -
3355729790*c3002000 + 136076220*c3000020 - 52490160*c3000002 +
670491000*c2300000*G3**2 + 536392800*c2300000*G3 + 67049100*c2300000 +
804589200*c2200001*G3 + 431412480*c2200001 + 134525880*c2030000*G2**2 -
177231240*c2030000*G2 - 185417100*c2030000 + 177231240*c2020010*G2 +
237156660*c2020010 - 335245500*c1400000* G3**2 - 134098200*c1400000*G3 -
536392800*c1300001*G3 - 186588360* c1300001 - 349481520*c1040000*G2**2 -
446283000*c1040000*G2 - 96801480 *c1040000 + 446283000*c1030010*G2 +
141533460*c1030010 + 67049100* c0500000*G3**2 - 1050269211*c0500000 +
3484701095*c0401000 + 134098200* c0400001*G3 + 22412970*c0310100 -
3498749570*c0302000 - 29656530* c0301100 + 59733720*c0301001 +
22412970*c0300200 + 81608040*c0300002 + 44825940*c0220100 - 179201160*c0201002
+ 22412970*c0130100 + 126340020* c0050000*G2**2 + 252680040*c0050000*G2 -
935561301*c0050000 + 3542861645*c0041000 - 252680040*c0040010*G2 -
252680040*c0040010 - 3496444550*c0032000 + 29656530*c0031100 +
22412970*c0030200 + 106537680* c0030020 + 2658766140*c0004100 +
824039730*c0004010)/533817540
```

53: c0201200 = (1288440*c5000000*G2**2 - 47455842*c5000000 + 4178520* c4010000*G2**2 - 2576880*c4010000*G2 + 157268345*c4001000 + 2576880* c4000010*G2 - 1426140*c3100001 - 29598480*c3020000*G2**2 - 10933920* c3020000*G2 + 1288440*c3020000 + 10933920*c3010010*G2 - 1150740* c3010010 - 147384260*c3002000 + 462780*c3000020 - 1426140*c3000002 - 2852280*c2200001 + 50839920*c2030000*G2**2 + 48263040*c2030000*G2 + 6755400*c2030000 - 48263040*c2020010*G2 - 18702360*c2020010 - 1426140* c1300001 - 36040680*c1040000*G2**2 - 53416800*c1040000*G2 - 17376120* c1040000 + 53416800*c1030010*G2 + 28134540*c1030010 - 48086379*c0500000 + 160421030*c0401000 + 3852630*c0310100 - 164300180*c0302000 + 3939840* c0301100 - 6366330*c0301001 + 3852630*c0300200 - 1426140*c0300002 + 7705260*c0220100 + 19098990*c0201002 + 3852630*c0130100 + 9332280* c0050000*G2**2 + 18664560*c0050000*G2 - 37352694*c0050000 + 153414005* c0041000 - 18664560*c0040010*G2 - 18664560*c0040010 - 150400700* c0032000 + 2765070*c0031100 + 3852630*c0030200 + 8506620*c0030020 - 49771260*c0011300 + 160374060*c0004100 + 10138095*c0004010)/20114730

54: c1001003 = (5749920*c5000000*G2**2 + 61904544*c5000000 - 21025440 *c4010000*G2**2 - 11499840*c4010000*G2 - 207784690*c4001000 + 11499840 *c4000010*G2 + 2121255*c3100001 + 26602560*c3020000*G2**2 + 30551040* c3020000*G2 + 5749920*c3020000 - 30551040*c3010010*G2 - 15275520* c3010010 + 210862120*c3002000 + 1654425*c3001001 + 5149440*c3000020 + 2121255*c3000002 + 4242510*c2200001 - 11154240*c2030000*G2**2 - 22654080*c2030000*G2 - 9525600*c2030000 + 22654080*c2020010*G2 + 15275520*c2020010 + 2121255*c1300001 - 2147040*c1040000*G2**2 - 345600 *c1040000*G2 + 1801440*c1040000 + 345600*c1030010*G2 - 3602880* c1030010 + 60984738*c0500000 - 203185660*c0401000 - 356535*c0310100 + 200364460*c0302000 - 3301155*c0301100 + 1536435*c0301001 - 356535* c0300200 + 2121255*c0300002 - 713070*c0220100 + 8833860*c0201200 - 9572580*c0201002 - 356535*c0130100 + 1974240*c0050000*G2**2 + 3948480* c0050000*G2 + 63516948*c0050000 - 205975510*c0041000 - 3948480*c0040010 *G2 - 3948480*c0040010 + 212278000*c0032000 + 356535*c0031100 - 356535* c0030200 + 1373760*c0030020 - 6417630*c0011300 - 75277620*c0004100 - 88583490*c0004010) /29779650

55: c1010030 = (9463392*c5000000*G2**2 - 45725472*c5000000 - 27897696 *c4010000*G2**2 - 18926784*c4010000*G2 + 152992306*c4001000 + 18926784 *c4000010*G2 - 251883*c3100001 + 16956864*c3020000*G2**2 + 36868608* c3020000*G2 + 9463392*c3020000 - 36868608*c3010010*G2 - 16152480* c3010010 - 154818472*c3002000 - 2281824*c3001010 - 240597*c3001001 + 10664352*c3000020 - 251883*c3000002 - 503766*c2200001 + 21881664* c2030000*G2**2 + 2954880*c2030000*G2 - 8970912*c2030000 - 2954880* c2020010*G2 + 22997952*c2020010 - 251883*c1300001 - 30360096*c1040000* G2**2 - 40808448*c1040000*G2 - 10448352*c1040000 + 40808448*c1030010* G2 + 23178528*c1030010 + 4330746*c1001003 - 44303442*c0500000 + 145882156 *c0401000 + 71307*c0310100 - 140412124*c0302000 + 405783*c0301100 - 225207*c0301001 + 71307*c0300200 - 251883*c0300002 + 142614*c0220100 - 1003428*c0201200 + 1397412*c0201002 + 71307*c0130100 + 9955872* c0050000*G2**2 + 19911744*c0050000*G2 - 35722404*c0050000 + 152756326* c0041000 - 19911744*c0040010*G2 - 19911744*c0040010 - 155003152* c0032000 - 71307*c0031100 - 2281824*c0031010 + 71307*c0030200 + 11156832 *c0030020 + 1283526*c0011300 + 9818820*c0004100 + 133400898*c0004010)/41072832

```
56:  c1002020 = (12960*c5000000*G2**2 - 85392*c5000000 - 38880*
c4010000*G2**2  - 25920*c4010000*G2 + 274190*c4001000 + 25920* c4000010*G2 -
2565*c3100001 + 25920*c3020000*G2**2 + 51840*c3020000* G2 + 12960*c3020000 -
51840*c3010010*G2 - 25920*c3010010 - 215480* c3002000 + 2565*c3001001 +
17280*c3000020 - 2565*c3000002 - 5130* c2200001 + 25920*c2030000*G2**2 -
12960*c2030000 + 25920*c2020010 - 2565*c1300001  - 38880*c1040000*G2**2 -
51840*c1040000*G2 - 12960* c1040000 + 51840*c1030010*G2 + 25920*c1030010 -
46170*c1001003 - 76206 *c0500000 + 228260*c0401000 + 2565*c0310100 -
150980*c0302000 + 12825* c0301100 - 12825*c0301001 + 2565*c0300200 -
2565*c0300002 + 5130* c0220100 - 30780*c0201200 + 30780*c0201002 +
2565*c0130100 + 12960* c0050000*G2**2  + 25920*c0050000*G2 - 84972*c0050000 +
336890* c0041000 - 25920*c0040010*G2 - 25920*c0040010 - 395600*c0032000 - 2565
*c0031100 + 2565*c0030200 + 17280*c0030020 + 46170*c0011300 - 1036260* c0004100
+ 1676430*c0004010)/492480
```

```
57:  c0210200 = ( - 285*c5000000 + 950*c4001000 - 950*c3002000 - 330*
c0500000 + 1175*c0401000 + 540*c0310100 - 1490*c0302000 - 486*c0301100 -
54*c0301001 + 540*c0220100 + 1458*c0201200 + 162*c0201002 - 285* c0050000 +
950*c0041000 - 950*c0032000 + 675*c0004010)/1620
```

```
58:  c0010040 = (2790*c5000000*G2**2 - 16673*c5000000 - 6660*c4010000 *G2**2
- 5580*c4010000*G2 + 55545*c4001000 + 5580*c4000010*G2 - 1260 *c3020000*G2**2 +
7740*c3020000*G2 + 2790*c3020000 - 7740*c3010010* G2 - 5010*c3010010 -
55450*c3002000 - 570*c3001010 + 3150*c3000020 + 15840*c2030000*G2**2 +
10260*c2030000*G2 - 1080*c2030000 - 10260* c2020010*G2 + 2160*c2020010 -
15210*c1040000*G2**2 - 21420*c1040000* G2 - 6210*c1040000 + 21420*c1030010*G2 +
11850*c1030010 - 5130* c1002020 - 16221*c0500000 + 53285*c0401000 -
50930*c0302000 + 4500* c0050000*G2**2 + 9000*c0050000*G2 - 12211*c0050000 +
55735*c0041000 - 9000*c0040010*G2 - 9000*c0040010 - 55830*c0032000 +
570*c0031010 + 6570*c0030020 - 1710*c0004100 + 59760*c0004010)/15390
```

```
59:  c0003200 = ( - 138*c5000000 + 455*c4001000 - 440*c3002000 - 141*
c0500000 + 470*c0401000 - 470*c0302000 - 141*c0050000 + 470*c0041000 -
470*c0032000 + 675*c0004100)/270
```

```
60:  c0010400 = ( - 1620*c5000000*G2**2 + 62259*c5000000 - 25920*
c4010000*G2**2 + 3240*c4010000*G2 - 207530*c4001000 - 3240*c4000010* G2 +
119880*c3020000*G2**2 + 55080*c3020000*G2 - 1620*c3020000 - 55080*c3010010*G2 +
3240*c3010010 + 207530*c3002000 - 2160*c3000020 - 187920*c2030000*G2**2 -
184680*c2030000*G2 - 29160*c2030000 + 184680 *c2020010*G2 + 78840*c2020010 +
127980*c1040000*G2**2 + 191160* c1040000*G2 + 63180*c1040000 -
191160*c1030010*G2 - 105840*c1030010 - 184680*c1010030 + 30780*c1002020  +
60726*c0500000 - 199865*c0401000 + 11970*c0310100 + 192200*c0302000 -
11970*c0301100 - 5130*c0300200 + 20520*c0220100 - 30780*c0210200 +
61560*c0201200 + 8550*c0130100 - 32400*c0050000*G2**2 - 64800*c0050000*G2 +
26040*c0050000 - 188435* c0041000 + 64800*c0040010*G2 + 64800*c0040010 +
162500*c0032000 - 8550 *c0031100 - 20520*c0031010 + 15390*c0030200 -
63720*c0030020 + 277020* c0011300 + 277020*c0010040 + 530955*c0004100 -
361530*c0004010 - 548910* c0003200)/184680
```

```
61:  c1001030 = (37980*c5000000*G2**2 - 240573*c5000000 - 87840*
c4010000*G2**2 - 75960*c4010000*G2 + 801910*c4001000 + 75960*c4000010 *G2 -
28440*c3020000*G2**2 + 99720*c3020000*G2 + 37980*c3020000 - 99720*c3010010*G2 -
75960*c3010010 - 801910*c3002000 + 43920*c3000020 + 232560*c2030000*G2**2 +
156600*c2030000*G2 - 11880*c2030000 - 156600*c2020010*G2 + 9720*c2020010 -
218340*c1040000*G2**2 - 308520* c1040000*G2 - 90180*c1040000 +
308520*c1030010*G2 + 166320*c1030010 + 126360*c1010030 - 5940*c1002020  -
234042*c0500000 + 769255*c0401000 - 8190*c0310100 - 736600*c0302000 +
8190*c0301100 + 3510*c0300200 - 14040 *c0220100 + 21060*c0210200 -
42120*c0201200 - 5850*c0130100 + 64080* c0050000*G2**2 + 128160*c0050000*G2 -
173880*c0050000 + 788845* c0041000 - 128160*c0040010*G2 - 128160*c0040010 -
771100*c0032000 + 5850*c0031100  + 14040*c0031010 - 10530*c0030200 +
106200*c0030020 - 189540*c0011300 + 126360*c0010400 - 325620*c0010040 -
363285*c0004100 + 915030*c0004010 + 375570*c0003200)/90720

62:  c1100003 = (23760*c5000000*G2**2 - 116817*c5000000 - 47520*
c4010000*G2**2 - 47520*c4010000*G2 + 398680*c4001000 + 47520*c4000010 *G2 +
15930*c3100001 - 47520*c3020000*G2**2 + 47520*c3020000*G2 + 23760*c3020000 -
47520*c3010010*G2 - 23760*c3010010  - 434470* c3002000 - 23760*c3001010 -
15930*c3001001 + 23760*c3000020 - 270* c3000002 + 31860*c2200001 +
190080*c2030000*G2**2 + 142560*c2030000* G2 - 142560*c2020010*G2 +
23760*c2020010 + 15930*c1300001 - 166320* c1040000*G2**2 - 237600*c1040000*G2 -
71280*c1040000 + 237600* c1030010*G2 + 142560*c1030010 - 213840*c1010030  -
71280*c1002020 + 213840*c1001030 + 140940*c1001003 - 112332*c0500000  +
376255*c0401000 + 8100*c0310100 - 384400*c0302000 - 8100*c0301100 -
15930*c0301001 - 270*c0300002 + 8100*c0220100 - 24300*c0210200 + 24300*c0201200
+ 46980 *c0201002 + 47520*c0050000*G2**2 + 95040*c0050000*G2 - 65076* c0050000
+ 377575*c0041000 - 95040*c0040010*G2 - 95040*c0040010 - 384340*c0032000 +
47520*c0030020 - 38205*c0004100 + 203580*c0004010 + 121770*c0003200)/145800

63:  c1000040 = (95040*c5000000*G2**2 + 116817*c5000000 - 332640*
c4010000*G2**2 - 190080*c4010000*G2 - 398680*c4001000 + 190080* c4000010*G2 -
15930*c3100001 + 380160*c3020000*G2**2 + 475200* c3020000*G2 + 95040*c3020000 -
475200*c(3010010*G2 - 213840*c3010010 + 434470*c3002000 + 23760*c3001010 +
15930*c3001001 + 166320*c3000020 + 270*c3000002 - 31860*c2200001 -
95040*c2030000*G2**2  - 285120* c2030000*G2 - 142560*c2030000 +
285120*c2020010*G2 + 261360*c2020010 - 15930*c1300001 + 145800*c1100003 -
95040*c1040000*G2**2 - 95040* c1040000*G2 + 95040*c1030010*G2 + 213840*c1010030
+ 213840*c1001030 - 140940*c1001003 + 112332*c0500000 - 376255*c0401000 -
8100*c0310100 + 384400*c0302000 + 8100*c0301100 + 15930*c0301001 + 270*c0300002
- 8100 *c0220100 + 24300*c0210200 - 24300*c0201200 - 46980*c0201002 + 47520*
c0050000*G2**2 + 95040*c0050000*G2 + 160116*c0050000 - 377575* c0041000 -
95040*c0040010*G2 - 95040*c0040010 + 384340*c0032000 + 47520 *c0030020 + 38205
*c0004100 - 203580*c0004010 - 121770*c0003200)/641520

64:  c0101300 = (34065*c5000000 - 113550*c4001000 + 113550*c3002000 +
34054*c0500000 - 113495*c0401000 - 330*c0310100 + 113440*c0302000 + 330
*c0301100 + 2970*c0300200 + 5940*c0210200 + 330*c0130100 + 34076* c0050000 -
113605*c0041000 + 113660*c0032000 - 330*c0031100 - 990* c0030200 -
66420*c0011300 + 35640*c0010400 - 334035*c0004100 + 990* c0004010 +
302310*c0003200)/48600
```

65: c0103100 = (45*c5000000 - 150*c4001000 + 150*c3002000 + 46* c0500000 - 155*c0401000 + 160*c0302000 + 44*c0050000 - 145*c0041000 + 140*c0032000 - 75*c0004100 - 90*c0004010 + 90*c0003200)/60

66: c0102200 = (261*c5000000 - 870*c4001000 + 870*c3002000 + 261* c0500000 - 870*c0401000 + 870*c0302000 + 22*c0300200 + 33*c0210200 - 333*c0101300 + 261*c0050000 - 870*c0041000 + 870*c0032000 - 11* c0030200 - 531*c0011300 + 297*c0010400 - 2538*c0004100 + 2311*c0003200)/33

67: c0022100 = (1161*c5000000 - 3870*c4001000 + 3870*c3002000 + 1161* c0500000 - 3870*c0401000 + 3870*c0302000 + 99*c0300200 + 11*c0220100 + 165*c0210200 + 33*c0201200 + 11*c0130100 - 66*c0102200 - 1611* c0101300 + 1161*c0050000 - 3870*c0041000 + 3870*c0032000 - 11*c0031100 - 33*c0030200 - 2205*c0011300 + 1188*c0010400 - 11259*c0004100 + 10206 *c0003200)/11

68: c0120200 = (- 129*c5000000 + 430*c4001000 - 430*c3002000 - 129* c0500000 + 430*c0401000 - 430*c0302000 - 11*c0300200 - 22*c0210200 + 11 *c0102200 + 168*c0101300 - 129*c0050000 + 430*c0041000 - 430*c0032000 + 234*c0011300 - 99*c0010400 + 1251*c0004100 - 1134*c0003200)/11

69: c2002010 = (88851*c5000000 - 300815*c4001000 - 7965*c3100001 + 318710*c3002000 + 7965*c3001001 + 135*c3000002 - 15930*c2200001 - 7965 *c1300001 + 72900*c1100003 - 70470*c1001003 + 87285*c0500000 - 292985* c0401000 - 4050*c0310100 + 300440*c0302000 + 4050*c0301100 + 7965* c0301001 + 135*c0300002 - 4050*c0220100 + 12150*c0210200 - 12150* c0201200 - 23490*c0201002 - 40590*c0103100 + 86064*c0050000 - 286880* c0041000 + 286880*c0032000 - 31635*c0004100 - 162675*c0004010)/11880

70: c2100002 = (- 77859*c5000000 + 262510*c4001000 + 3510*c3100001 - 273340 *c3002000 - 3510*c3001001 - 540*c3000002 + 5670*c2200001 + 5670 *c2002010 + 2160*c1300001 - 24300*c1100003 + 38880*c1001003 - 76770* c0500000 + 257065*c0401000 - 260560*c0302000 - 2160*c0301001 - 540* c0300002 + 8910*c0201002 + 46260*c0103100 - 76071*c0050000 + 253570* c0041000 - 253570*c0032000 + 2340*c0004100 + 166725*c0004010)/4050

71: c0003020 = (- 276*c5000000 + 920*c4001000 - 920*c3002000 - 276* c0500000 + 920*c0401000 - 920*c0302000 + 180*c0103100 - 273*c0050000 + 905*c0041000 - 890*c0032000 + 225*c0004100 + 945*c0004010 - 270* c0003200)/270

72: c1002002 = (3093*c5000000 - 9106*c4001000 + 1566*c3100001 + 4972* c3002000 - 1566*c3001001 - 540*c3000002 + 1566*c2200001 - 4698*c2100002 + 1566*c2002010 - 4860*c1100003 + 23328*c1001003 + 3354*c0500000 - 10411*c0401000 + 8104*c0302000 - 540*c0300002 + 3078*c0201002 + 3354* c0050000 - 10411*c0041000 + 8104*c0032000 - 45657*c0004100 - 49572* c0004010 + 41526*c0003200 + 41526*c0003020)/7776

73: c0101003 = (- 573*c5000000 + 1574*c4001000 - 54*c3100001 - 548* c3002000 + 54*c3001001 + 162*c3000002 - 54*c2200001 + 162*c2100002 - 54 *c2002010 + 1458*c1100003 - 162*c1002002 - 2916*c1001003 - 582* c0500000 + 1619*c0401000 - 656*c0302000 + 162*c0300002 + 324*c0201002 - 582*c0050000 + 1619*c0041000 - 656*c0032000 + 18819*c0004100 + 18954* c0004010 - 17334*c0003200 - 17334*c0003020)/3402

74: c0202100 = (1860*c5000000 - 6329*c4001000 + 6716*c3002000 - 594*c2200001 - 1782*c2100002 - 594*c1300001 + 5346*c1100003 - 1782*c1002002 + 486*c1001003 + 1959*c0500000 - 6824*c0401000 + 7904*c0302000 + 594* c0301001 - 4860*c0101003 + 1860*c0050000 - 6329*c0041000 + 6716* c0032000 + 5832*c0004100 + 4347*c0004010 - 6966*c0003200 - 6966* c0003020)/594

75: c2001002 = (8889*c5000000 - 26228*c4001000 + 16022*c3002000 - 1782 *c3000002 - 594*c2200001 - 1782*c2100002 - 594*c1300001 - 10692* c1100003 + 32562*c1001003 + 8988*c0500000 - 26723*c0401000 + 17210* c0302000 + 594*c0301001 - 1782*c0300002 - 594*c0202100 - 3564*c0201002 + 32562*c0101003 + 8889*c0050000 - 26228*c0041000 + 16022*c0032000 - 202662*c0004100 - 204147*c0004010 + 183708*c0003200 + 183708*c0003020)/1782

76: c0110300 = (- 129*c5000000 + 430*c4001000 - 430*c3002000 - 129*c0500000 + 430*c0401000 - 430*c0302000 - 11*c0300200 - 11*c0210200 - 11 *c0201200 + 201*c0101300 - 129*c0050000 + 430*c0041000 - 430*c0032000 + 234*c0011300 - 99*c0010400 + 1251*c0004100 - 1134*c0003200)/33

77: c0002030 = (33*c5000000 - 110*c4001000 + 110*c3002000 + 9* c1000040 + 33 *c0500000 - 110*c0401000 + 110*c0302000 + 33*c0050000 - 110*c0041000 + 110*c0032000 + 9*c0010040 - 369*c0004010 + 486* c0003020)/234

78: c0100004 = (51*c5000000 - 149*c4001000 + 86*c3002000 - 11* c3000002 - 11 *c2100002 - 11*c2001002 - 33*c1100003 + 201*c1001003 + 51 *c0500000 - 149*c0401000 + 86*c0302000 + 234*c0101003 + 51*c0050000 - 149*c0041000 + 86*c0032000 - 1251*c0004100 - 1251*c0004010 + 1134* c0003200 + 1134*c0003020) /99

79: c0002300 = (3*c5000000 - 10*c4001000 + 10*c3002000 + 3*c0500000 - 10*c0401000 + 10*c0302000 + 3*c0101300 + 3*c0050000 - 10*c0041000 + 10*c0032000 + 3*c0011300 - 36*c0004100 + 54*c0003200)/33

80: c0011030 = (- 3*c5000000 + 10*c4001000 - 10*c3002000 - 3* c1001030 - 3 *c0500000 + 10*c0401000 - 10*c0302000 - 3*c0050000 + 10* c0041000 - 10*c0032000 + 36*c0004010 - 54*c0003020 + 33*c0002030)/3

81: c0002003 = (18*c5000000 - 61*c4001000 + 64*c3002000 + 3*c1001003 + 18*c0500000 - 61*c0401000 + 64*c0302000 + 3*c0101003 + 18*c0050000 - 61*c0041000 + 64*c0032000 + 36*c0004100 + 36*c0004010 - 54*c0003200 - 54*c0003020)/33

82: c0012200 = - c0102200 - c0003200 + 3*c0002300

83: c0012020 = - c1002020 - c0003020 + 3*c0002030

84: c0102002 = (- 17*c5000000 + 57*c4001000 - 58*c3002000 - 18* c1002002 - 17*c0500000 + 57*c0401000 - 58*c0302000 - 17*c0050000 + 57* c0041000 - 58*c0032000 + 18*c0003200 + 18*c0003020 + 54*c0002003)/18

85: c0013100 = - c0103100 - c0004100 + 3*c0003200

86: c1003010 = (276*c5000000 - 920*c4001000 + 920*c3002000 + 276* c0500000 - 920*c0401000 + 920*c0302000 + 273*c0050000 - 905*c0041000 + 890*c0032000 + 90*c0013100 - 495*c0004100 - 585*c0004010 + 270* c0003020)/90

87: c0003002 = (531*c5000000 - 1775*c4001000 + 1790*c3002000 - 90* c1003010 + 531*c0500000 - 1775*c0401000 + 1790*c0302000 - 90*c0103100 + 528*c0050000 - 1760*c0041000 + 1760*c0032000 - 585*c0004100 - 585* c0004010)/270

88: c0004001 = (3*c5000000 - 10*c4001000 + 10*c3002000 + 3*c0500000 - 10*c0401000 + 10*c0302000 + 3*c0050000 - 10*c0041000 + 10*c0032000 - 3*c0004100 - 3*c0004010)/3

89: c1004000 = (- 15*c5000000 + 50*c4001000 - 50*c3002000 - 15* c0500000 + 50*c0401000 - 50*c0302000 - 15*c0050000 + 50*c0041000 - 50* c0032000 + 27*c0004010 + 27*c0004001)/9

90: c0104000 = (- 3*c5000000 + 10*c4001000 - 10*c3002000 - 9* c1004000 - 3 *c0500000 + 10*c0401000 - 10*c0302000 - 3*c0050000 + 10* c0041000 - 10*c0032000 + 27*c0004001)/9

91: c0005000 = (3*c5000000 - 10*c4001000 + 10*c3002000 + 3*c0500000 - 10*c0401000 + 10*c0302000 + 3*c0050000 - 10*c0041000 + 10*c0032000)/9

92: c1003001 = (c5000000 - 5*c4001000 + 10*c3002000 + 55*c1004000 - 30 *c1003010 - c0005000)/30

93: c0103001 = (c0500000 - 5*c0401000 + 10*c0302000 + 55*c0104000 - 30 *c0103100 - c0005000)/30

94: c0013010 = (33*c5000000 - 110*c4001000 + 110*c3002000 - 55* c1004000 + 33*c0500000 - 110*c0401000 + 110*c0302000 - 55*c0104000 + 34 *c0050000 - 115*c0041000 + 120*c0032000 - 30*c0013100 + 65*c0005000)/30

95: c2002001 = (c5000000 - 5*c4001000 + 12*c3002000 - 6*c2002010 + 5* c1004000 - c0005000)/6

96: c0202001 = (c0500000 - 5*c0401000 + 12*c0302000 - 6*c0202100 + 5* c0104000 - c0005000)/6

97: c0022010 = (3*c5000000 - 10*c4001000 + 10*c3002000 - 5*c1004000 + 3*c0500000 - 10*c0401000 + 10*c0302000 - 5*c0104000 + 4*c0050000 - 15*c0041000 + 22*c0032000 - 6*c0022100 + 5*c0005000)/6

98: c0100400 = (c0300200 + c0210200 + c0201200 + 3*c0110300 + 3* c0101300) /9

99: c0021200 = - c0120200 - 3*c0110300 - c0030200 - 3*c0011300 + 9* c0010400

100: c0200300 = (c0300200 + c0210200 + c0201200)/3

101: c0020300 = (c0120200 + c0030200 + c0021200)/3

102: c2001020 = (- 2*c5000000*G2**2 + 7*c4010000*G2**2 + 4* c4010000*G2 - 4*c4000010*G2 - 8*c3020000*G2**2 - 10*c3020000*G2 - 2*c3020000 + 10*c3010010*G2 + 4*c3010010 - 5*c3000020 + 2*c2030000 *G2**2 + 6*c2030000*G2 + 3*c2030000 - 6*c2020010*G2 - 6*c2020010 + 2*c1040000*G2**2 + 2*c1040000*G2 - 2*c1030010*G2 - 9*c1010030 - 9 *c1001030 + 27*c1000040 - c0050000*G2**2 - 2*c0050000*G2 - c0050000 + 2*c0040010*G2 + 2*c0040010 - c0030020)/3

103: c0021020 = (- c5000000*G2**2 + 2*c4010000*G2**2 + 2*c4010000* G2 - 2 *c4000010*G2 + 2*c3020000*G2**2 - 2*c3020000*G2 - c3020000 + 2*c3010010*G2 + 2*c3010010 - c3000020 - 8*c2030000*G2**2 - 6* c2030000*G2 + 6*c2020010*G2 + 7*c1040000*G2**2 + 10*c1040000*G2 + 3*c1040000 - 10*c1030010*G2 - 6*c1030010 - 9*c1010030 - 2*c0050000 *G2**2 - 4*c0050000*G2 - 2*c0050000 + 4*c0040010*G2 + 4*c0040010 - 5*c0030020 - 9*c0011030 + 27*c0010040)/3

104: c2000030 = (2*c5000000*G2**2 - 7*c4010000*G2**2 - 4*c4010000* G2 + 4 *c4000010*G2 + 8*c3020000*G2**2 + 10*c3020000*G2 + 2* c3020000 - 10*c3010010*G2 - 4*c3010010 + 5*c3000020 - 2*c2030000* G2**2 - 6*c2030000*G2 - 3*c2030000 + 6*c2020010*G2 + 6*c2020010 + 3 *c2001020 - 2*c1040000*G2**2 - 2*c1040000*G2 + 2*c1030010*G2 + c0050000*G2**2 + 2*c0050000*G2 + c0050000 - 2*c0040010*G2 - 2* c0040010 + c0030020)/9

105: c0020030 = (c5000000*G2**2 - 2*c4010000*G2**2 - 2*c4010000*G2 + 2*c4000010*G2 - 2*c3020000*G2**2 + 2*c3020000*G2 + c3020000 - 2* c3010010*G2 - 2*c3010010 + c3000020 + 8*c2030000*G2**2 + 6*c2030000* G2 - 6*c2020010*G2 - 7*c1040000*G2**2 - 10*c1040000*G2 - 3* c1040000 + 10*c1030010*G2 + 6*c1030010 + 2*c0050000*G2**2 + 4* c0050000*G2 + 2*c0050000 - 4*c0040010*G2 - 4*c0040010 + 5*c0030020 + 3*c0021020)/9

106: c1000004 = (c3000002 + c2100002 + c2001002 + 3*c1100003 + 3* c1001003) /9

107: c1200002 = - 3*c1100003 - c0300002 - c0201002 - 3*c0101003 + 9* c0100004

108: c2000003 = (c3000002 + c2100002 + c2001002)/3

109: c0200003 = (c1200002 + c0300002 + c0201002)/3

110: c0014000 = (3*c5000000 - 10*c4001000 + 10*c3002000 - 5*c1004000 + 3*c0500000 - 10*c0401000 + 10*c0302000 - 5*c0104000 + 3*c0050000 - 10*c0041000 + 10*c0032000 + 6*c0005000)/5

111: c0001040 = - c0005000 + 4*c0004010 - 6*c0003020 + 4*c0002030

112: c0001400 = - c0005000 + 4*c0004100 - 6*c0003200 + 4*c0002300

113: c0000005 = - 4*c0005000 + 15*c0004001 - 20*c0003002 + 10* c0002003

114: c2003000 = (c5000000 - 5*c4001000 + 10*c3002000 + 5*c1004000 - c0005000) /10

115: c0000050 = c0005000 - 5*c0004010 + 10*c0003020 - 10*c0002030 + 5* c0001040

```
116:  c0023000 = (c0050000 - 5*c0041000 + 10*c0032000 + 5*c0014000 - c0005000)
/10

117:  c0000500 = c0005000 - 5*c0004100 + 10*c0003200 - 10*c0002300 + 5*
c0001400

118:  c0203000 = (c0500000 - 5*c0401000 + 10*c0302000 + 5*c0104000 - c0005000)
/10

119:  c0001004 = ( - c0005000 + 5*c0004001 - 10*c0003002 + 10*c0002003 +
c0000005)/5

120:  c1020020 = (c5000000*G2**2 - 2*c4010000*G2**2 - 2*c4010000*G2 +
2*c4000010*G2 - 2*c3020000*G2**2 + 2*c3020000*G2 + c3020000 - 2* c3010010*G2 -
2*c3010010 + c3000020 + 8*c2030000*G2**2 + 6*c2030000* G2 - 6*c2020010*G2 -
7*c1040000*G2**2 - 10*c1040000*G2 - 3* c1040000 + 10*c1030010*G2 + 6*c1030010 +
2*c0050000*G2**2 + 4* c0050000*G2 + 2*c0050000 - 4*c0040010*G2 - 4*c0040010 +
2*c0030020)/3

121:  c2010020 = (c5000000*G2**2 - 5*c4010000*G2**2 - 2*c4010000*G2 +
2*c4000010*G2 + 10*c3020000*G2**2 + 8*c3020000*G2 + c3020000 - 8 *c3010010*G2 -
2*c3010010 + c3000020 - 10*c2030000*G2**2 - 12* c2030000*G2 - 3*c2030000 +
12*c2020010*G2 + 6*c2020010 + 5*c1040000 *G2**2 + 8*c1040000*G2 + 3*c1040000 -
8*c1030010*G2 - 6*c1030010 + 3*c1020020 - c0050000*G2**2 - 2*c0050000*G2 -
c0050000 + 2*c0040010* G2 + 2*c0040010 - c0030020)/3
```

PA/ed

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
2620

2. GOVT ACCESSION NO.
AD-A137930

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
A BIVARIATE C^2 CLOUGH-TOCHER SCHEME

5. TYPE OF REPORT & PERIOD COVERED
Summary Report - no specific reporting period

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Peter Alfeld

8. CONTRACT OR GRANT NUMBER(s)
DAAG29-80-C-0041
DE-AC02-82ER12046

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Mathematics Research Center, University of Wisconsin
610 Walnut Street
Madison, Wisconsin 53706

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Work Unit Number 3 - Numerical Analysis and Scientific Computing

11. CONTROLLING OFFICE NAME AND ADDRESS
(See Item 18 below)

12. REPORT DATE
January 1984

13. NUMBER OF PAGES
33

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES
U. S. Army Research Office
P. O. Box 12211
Research Triangle Park
North Carolina 27709

U. S. Department of Energy
Washington, DC 20545

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Bivariate interpolation, Clough-Tocher schemes

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
A Clough-Tocher like interpolation scheme is described for values of position, gradient and Hessian at scattered points in two variables. The domain is assumed to have been triangulated. The interpolant has local support, is globally twice differentiable, piecewise polynomial, and reproduces polynomials of degree up to three exactly.

END

FILMED

3-84

DTIC